
Droit des logiciels

...ou du moins un petit aperçu !

François Pellegrini
Professeur, Université de Bordeaux
francois.pellegrini@labri.fr

Ce document est copiable et distribuable librement et gratuitement à la condition expresse que son contenu ne soit modifié en aucune façon, et en particulier que le nom de son auteur et de son institution d'origine continuent à y figurer, de même que le présent texte.

Quelques notions d'économie des biens immatériels

Copier n'est pas voler...

Économie des biens immatériels (1)

- L'économie des biens immatériels diffère fondamentalement de l'économie des biens matériels
- On ne peut pas penser la révolution numérique et les changements profonds qu'elle induit si l'on reste prisonnier des schémas de pensée du millénaire précédent

Économie des biens immatériels (2)

- À la différence des biens matériels, les biens immatériels sont non rivaux
 - On n'est pas en rivalité pour les consommer
 - On peut les partager sans s'appauvrir
 - On ne « donne » pas une idée, on la copie !
- La notion de « vol » n'est pas pertinente
 - Pas de « propriétaire »
 - Le terme « propriété intellectuelle » est intellectuellement erroné
 - C'est un oxymore
 - Différentes incriminations : contrefaçon, détournement de finalité, parasitisme, etc.

Économie des biens immatériels (3)

- Le coût marginal des biens numériques est nul
 - Coût marginal : coût de copie d'une unité supplémentaire d'un bien
 - Bien matériel : voiture : prototype → série
 - Bien immatériel : copie à coût marginal nul
 - Mais ordinateur et électricité sont des biens rivaux
- Un bien numérique (logiciel, livre, musique) peut être distribué gratuitement dès le moment où son développement a été financé
- À la différence des services à entretenir

Économie des biens immatériels (4)

- Les effets de réseau sont considérables
 - La valeur d'un produit augmente avec le nombre de personnes qui l'utilisent
 - Ex. : téléphone
- Très grande volatilité du marché
- Obsolescence très rapide
- Un logiciel non utilisé est un logiciel qui meurt

Valeur des logiciels

- Tout logiciel a un coût
 - Moyens mis en œuvre pour le produire
 - Facile à quantifier
- Tout logiciel a une valeur
 - Valeur d'usage
 - Découle du service qu'il rend
 - Valeur intrinsèque
 - Expertise contenue au sein du code source
 - Maintenabilité, extensibilité, réutilisabilité
 - Très difficile à quantifier !
 - Décorrélée du coût de production

Quelques définitions

Informatique

- L'informatique est la science du traitement efficace de l'information
- L'informatique n'est pas la « science des ordinateurs » !
 - « L'informatique n'est pas plus la science des ordinateurs que l'astronomie n'est la science des télescopes »
 - Le terme « *computer science* » est erroné

Langue

- « Système permettant de communiquer une pensée »
- Les langues écrites communiquent cette pensée sous forme de textes
- Une langue écrite est définie par :
 - Un lexique : ensemble de mots admis par la langue
 - Une grammaire : ensemble de règles permettant d'assembler et d'agencer les mots de la langue
- Une langue n'est pas un texte
- Elle sert à écrire des textes

Langage informatique (1)

- Les langues humaines sont ambiguës :
 - « Elle prend la boule et la lance »
 - « *Time flies like an arrow* »
 - « *A veces me siento solo ; otras me ayudan a sentarme* »
 - Etc.
- Les langues / langages informatiques ont été conçus pour être univoques : un texte n'a qu'un seul sens
 - La « théorie des langages » permet de le prouver

Langage informatique (2)

- Un langage informatique est aussi une langue de communication humaine
 - Un programmeur, en lisant le programme d'un confrère, peut en comprendre le sens voire y détecter et corriger les erreurs (« bogues »)
- Il existe de nombreux langages informatiques, possédant chacun un mode de pensée ou d'expression spécifiques
 - Exemples : C, Python, Java, Ruby, HTML, etc.
 - Analogues aux différents jargons professionnels

Donnée

- « Élément d'information ayant un sens pour celui qui l'a produit »
- « Description élémentaire d'une réalité »
- Lorsque plusieurs données doivent être traitées ensemble, on peut les organiser au moyen d'une structure de données
 - Exemple : Position = (Latitude, Longitude)
 - Agrégation de types simples définie par une grammaire
 - Appelée « type abstrait » en informatique
 - Format de données + règles de traitement

Algorithme

- « Ensemble d'étapes permettant d'arriver à un résultat donné à partir d'éléments fournis »
- Utilisés dans de nombreux domaines :
 - Recette de cuisine
 - La quiche aux lardons
 - Trame d'histoire
 - Le Petit chaperon rouge
 - Méthode mathématique
 - Calcul du PGCD
- Appartiennent au domaine des idées
- Appartiennent au fonds commun

Programme d'ordinateur

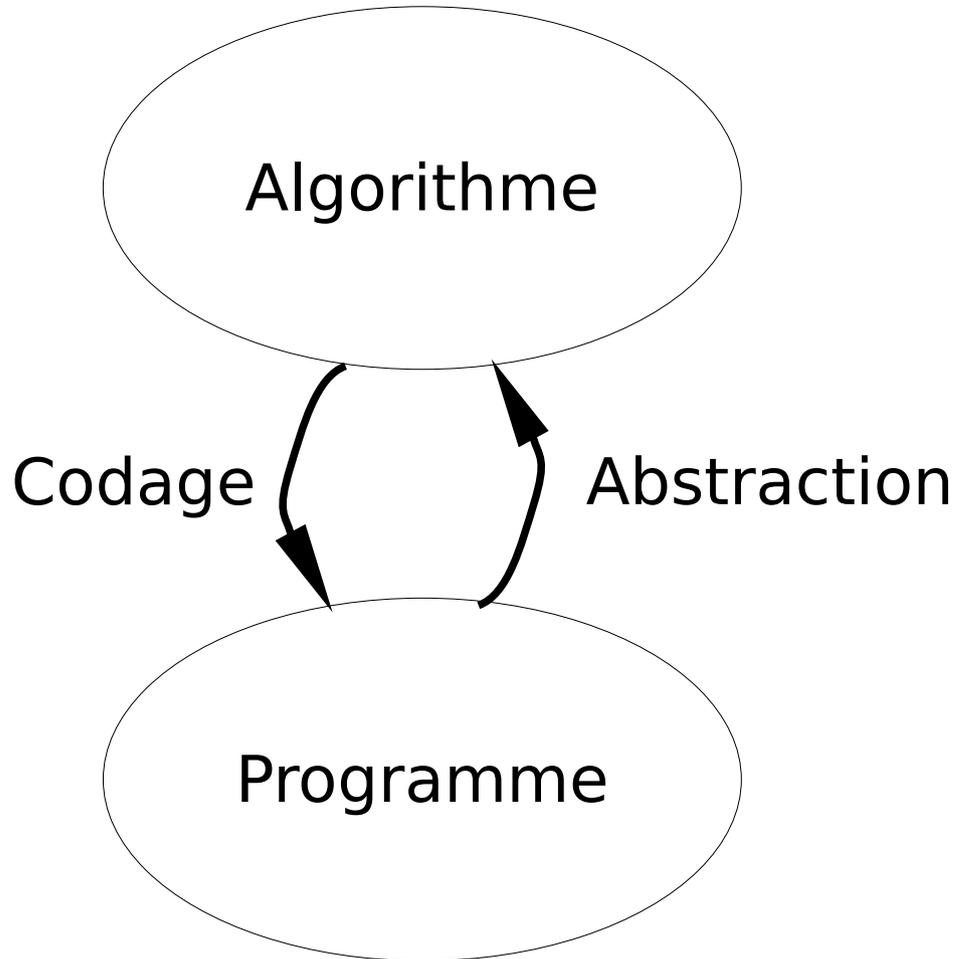
« Expression, dans un ou plusieurs langages de programmation, d'un ensemble d'algorithmes visant à la réalisation d'une tâche intellectuelle donnée »

- Toute communication textuelle d'un algorithme est réductible à un programme
 - Dans un langage informatique ou humain
 - Les logigrammes (organigrammes) sont des expressions mixtes graphiques / textuelles
 - Peu utilisées car très peu expressives

Logiciel

- « Logiciel » et « programme d'ordinateur » sont des termes équivalents
- Exprime la nature « logique » du programme
 - Opposition « matériel » / « logiciel » analogue à l'opposition « hardware » / « software »

Algorithmes et programmes



- Algorithmes :
 - Des idées
 - Des mathématiques
- Programmes :
 - Des œuvres de l'esprit
 - Du discours
 - Humain → humain
 - Humain → ordinateur
- Traitements :
 - Lorsque exécutés

- Analogue au processus de création littéraire

Histoire du logiciel

Et le logiciel fut !

Apparition du logiciel

- Les premiers calculateurs ne possédaient pas de logiciel
 - Programmés « à la main »
- Le logiciel est apparu avec l'idée que le programme pouvait être stocké dans la même mémoire que celle utilisée pour les données
 - Architecture dite « de von Neumann »
 - Le logiciel est une donnée
 - Mais pas comme les autres !
 - Possibilité de programmes auto-modifiables
 - Méta-programmation, compilation

Apparition du droit du logiciel (1)

- Tant que les logiciels étaient spécifiques à un ordinateur donné, la question de leur statut ne se posait pas
 - Les logiciels étaient fournis « gratuitement » aux utilisateurs, en tant que « fournitures annexes »
 - Au même titre que les manuels d'utilisation

Apparition du droit du logiciel (2)

- Les fournisseurs encourageaient leurs clients à modifier les logiciels et à partager les modifications
- Soutien à la création de « clubs d'utilisateurs » servant à l'échange de ces améliorations
 - Mutualisation et donc réduction des coûts de maintenance logicielle, offrant un avantage compétitif sur les concurrents
 - Les principes du logiciel libre avant l'heure !

Apparition du droit du logiciel (3)

- La question du statut juridique du logiciel s'est posée lorsque sont apparus les premiers ordinateurs compatibles avec les grands systèmes IBM, à la fin des années 1960
 - Les clients de matériels compatibles utilisaient les logiciels d'IBM sans payer
- IBM décida donc de facturer séparément logiciel et matériel
 - Politique de « dégroupage » (« *unbundling* »)
 - Mais comment caractériser juridiquement le logiciel ?

Quel statut pour le logiciel ? (1)

- Trois voies offertes au législateur :
 - Droit des brevets
 - Jugé inadapté de par l'objet à réguler et la lourdeur des mécanismes de dépôt et d'entretien
 - Droit « sui generis »
 - Taillé sur mesure
 - Durée de mise en œuvre des conventions internationales
 - Droit d'auteur
 - Processus de création similaire entre logiciels et œuvres littéraires
 - Existence de la Convention de Berne (1886)
 - Protection internationale automatique et immédiate

Quel statut pour le logiciel ? (2)

- C'est le droit d'auteur qui fut choisi comme véhicule du droit du logiciel
 - 1980 aux États-Unis (modification du Copyright Act)
 - 1985 en France (loi du 3 juillet 1985)
 - 1991 au sein de l'Union européenne (directive 91/250/CE)
 - 1994 parmi les membres de l'OMC (accords ADPIC)
 - 1996 au niveau mondial (traité WCT de l'OMPI)

Quel statut pour les langages ? (1)

- Une langue n'est pas une œuvre
 - Elle permet d'écrire des œuvres
 - Elle est d'un niveau d'abstraction supérieur
 - C'est un système de pensée et non une création de forme exprimant une pensée
 - Pas de revendication possible sous le régime du droit d'auteur
- Un langage informatique est aussi une langue de communication humaine
 - Permet l'échange d'informations entre humains

Quel statut pour les langages ? (2)

- Les langages informatiques ne peuvent donc être appropriés sans porter préjudice à des droits plus élevés
- Un format de fichiers définit la grammaire d'un langage
 - Accorder des brevets sur des formats de fichiers reviendrait donc à pouvoir monopoliser un langage
 - Donc aussi par extension des éléments de langues humaines

Le droit d'auteur

...ou, plus exactement, un bref aperçu
d'icelui

Droit des biens immatériels

- Définis en France dans le « Code de la Propriété Intellectuelle » (CPI)
 - Terme « impropre », car c'est un oxymore
- Organisation hiérarchique du Code :
 - Code de la propriété littéraire et artistique
 - Droits d'auteur
 - Droits « voisins des droits de l'auteur »
 - Droits des interprètes
 - Code de la propriété industrielle
 - Droit des brevets
 - Droit des marques

Justification du droit d'auteur (1)

- Encourager globalement la création en garantissant aux créateurs un monopole temporaire sur l'exploitation de leurs créations
- Au terme de ce monopole, les créations gagnent le domaine public, et peuvent ainsi bénéficier à tous
 - Une œuvre ne « tombe » pas dans le domaine public, elle s'y élève !

Justification du droit d'auteur (2)

- Le monopole temporaire concédé au titre du droit d'auteur représente un compromis entre l'intérêt des auteurs et l'intérêt du public

« Le livre, comme livre, appartient à l'auteur, mais comme pensée, il appartient [...] au genre humain. [...] Si l'un des deux droits, le droit de l'écrivain et le droit de l'esprit humain, devait être sacrifié, ce serait, certes, le droit de l'écrivain, car l'intérêt public est notre préoccupation unique, et tous [...] doivent passer avant nous. »

**Victor Hugo,
Discours d'ouverture du congrès littéraire, 1878**

Structuration du droit d'auteur

- Organisation hiérarchique des droits d'auteur :
 - Droits patrimoniaux
 - Attachés à l'œuvre
 - Droits extra-patrimoniaux, aussi appelés droits moraux
 - Attachés à la personne de l'auteur (ou à ses descendants)
 - N'existent pas dans le système du copyright

Œuvre de l'esprit (1)

- L'œuvre est une création de forme
 - C'est la forme qui sera protégée, et non les idées et les concepts
- « Les idées sont de libre parcours » !
(H. Desbois)

Œuvre de l'esprit (2)

- Liste non limitative des œuvres susceptibles de recevoir la protection du droit d'auteur :
 - « Notamment » (L.112-2 CPI) : les livres, brochures et autres écrits littéraires, artistiques et scientifiques ; les conférences, allocutions, sermons, plaidoiries et autres œuvres de même nature ; les œuvres dramatiques ou dramatico-musicales ; les œuvres chorégraphiques, les numéros et tours de cirque, les pantomimes, etc.

Critère de protection

- Notion d'« originalité », reflétant la « personnalité de l'auteur »
 - Reformulé sous le terme d'« apport intellectuel » lorsque sera abordée la question des œuvres logicielles
- Une création de forme non originale ne sera pas susceptible de protection :
 - Photographie conforme d'un tableau ancien
 - Mais une photo floue, si !
 - Photos « non artistiques » de plats de cuisine
 - Photo prise par un animal non-humain

Critères non pertinents (1)

- La « nouveauté »
 - La nouveauté relève du domaine des idées, pas des créations de forme
 - Ne concerne pas le droit d'auteur
 - « Encore une histoire de mousquetaires ?! »
 - On peut créer de façon originale sans nouveauté
 - Des navigateurs web différents sont bien chacun des œuvres originales
 - Alors qu'ils doivent respecter des normes de comportement très contraignantes

Critères non pertinents (2)

- Le « mérite »
 - Le mérite relève de la morale, pas du droit
 - Qui se souvient encore des peintres « officiels » de la fin du XIXe siècle, par rapport aux « refusés » ?

Créations échappant à la protection

- Une création de forme ne reflétant pas la personnalité de son concepteur ne sera pas éligible à la protection par le droit d'auteur
 - Tables mathématiques, annuaires des marées, etc.
- Cependant, leur présentation graphique pourra être éligible si elle reflète la personnalité de son auteur (logos, décorations, etc.)

Domaine public initial

- Certaines œuvres échappent dès leur origine au droit d'auteur
 - C'est le cas des textes de loi
 - Textes « à portée normative »
 - Pas cependant de certaines normes techniques !

Fonds commun

- Le fonds commun contient l'ensemble des connaissances accessibles au public
 - N'est pas relatif aux œuvres, mais aux « archétypes », aux « algorithmes du monde réel »
 - À ne pas confondre avec le domaine public des œuvres de l'esprit
- Les algorithmes mathématiques utilisés en informatique appartiennent au fonds commun

Synthèse des concepts juridiques

- Concepts juridiques de la création intellectuelle

Fonds commun : entités abstraites

- algorithmes, spécifications fonctionnelles et d'interfaces, etc.

Créations de forme

Créations non originales :

- tables de marées,
mathématiques,
mise en œuvre
d'interfaces,
etc.

Œuvres de l'esprit

Entrées/placées
dans le domaine
public

Couvertes par
des droits
patrimoniaux
(licences libres
et privatives)

Automaticité de la protection (1)

- La protection est réputée acquise dès la conception de l'œuvre
 - L.111-1 CPI : « *L'auteur d'une œuvre de l'esprit jouit sur cette œuvre, du seul fait de sa création, d'un droit de propriété incorporelle exclusif et opposable à tous* »
 - L.111-2 CPI : « *L'œuvre est réputée créée, indépendamment de toute divulgation publique, du seul fait de la réalisation, même inachevée, de la conception de l'auteur* »

Automaticité de la protection (2)

- Aucune formalité d'enregistrement n'est nécessaire
- Il est prudent de se pré-constituer des preuves de paternité et d'antériorité :
 - Constat d'huissier
 - Dépôt chez un notaire
 - Dépôt auprès d'associations spécialisées
 - SGDL, APP, etc.
 - Courrier envoyé à soi-même
 - Publication de condensats comme preuve de possession
 - Etc.

Droits patrimoniaux

- Matérialisation du droit qu'a l'auteur de tirer profit de l'utilisation de son œuvre
 - Protection économique de l'œuvre
- Distincts de la possession physique de l'œuvre
 - On n'achète que la « matière » du tableau, pas ce qu'il représente
 - La cession des droits de reproduction doit être explicite
- Les droits patrimoniaux sont :
 - Cessibles
 - Temporaires

« Exceptions » aux droits de l'auteur

- Définies par la loi
 - Ce ne sont donc pas des exceptions individuelles mais un droit !
- Liste exhaustive incluant :
 - Copie privée à l'usage du copiste
 - Représentation dans le cercle de famille
 - Citation
 - Parodie
 - etc.

Durée des droits patrimoniaux

- La durée des droits patrimoniaux dépend des pays et des situations
 - Mais uniformisation mondiale sous la pression de l'industrie du divertissement
- Actuellement, au sein de l'Union européenne, cette durée est de 70 ans après le décès de l'auteur
 - Délai peu compatible avec la préservation du patrimoine logiciel

Droits extra-patrimoniaux

- Protègent tant l'œuvre que l'acte de création
- Trois droits distincts :
 - Droit à la paternité
 - Les contrats d'« auteur caché » (« *ghostwriter* ») sont illégaux et jugés comme tels
 - Droit au respect de l'œuvre
 - Droit spécifique pour le nom de l'œuvre
 - Droit de divulgation
- Les droits extra-patrimoniaux sont :
 - Inaliénables
 - Perpétuels

Personnes concernées

- En droit d'auteur, on distingue :
 - L'auteur : personne physique qui a réalisé la création de forme
 - Le titulaire de droits : personne physique ou morale qui détient certaines catégories de droits sur l'œuvre
 - Parfois appelée « ayant droit »
- Pas de « propriétaire » !

L'œuvre réalisée à plusieurs (1)

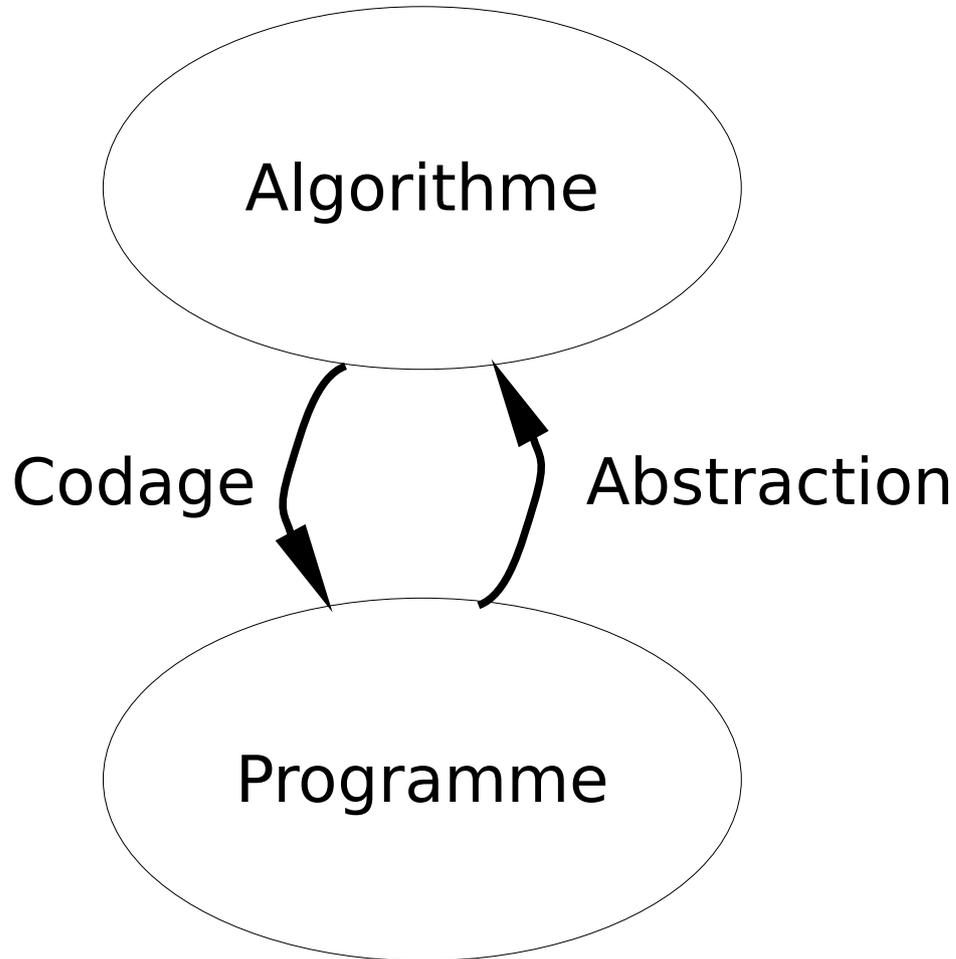
- Trois catégories juridiques (L.113-2 CPI)
 - Œuvre de collaboration
 - Œuvre à la réalisation de laquelle ont concouru plusieurs personnes physiques
 - Droits partagés
 - Œuvre composite (aussi appelée « œuvre dérivée »)
 - Œuvre nouvelle à laquelle est incorporée une œuvre préexistante sans la collaboration des auteurs de celle-ci
 - Exploitable sans préjudice des droits attachés aux œuvres incorporées

L'œuvre réalisée à plusieurs (2)

- Œuvre collective
 - Œuvre créée à l'initiative d'une personne physique ou morale qui l'édite, dans laquelle la contribution des différents auteurs se fond [...]
 - Exercice des droits patrimoniaux par la personne éditrice

Le droit d'auteur adapté au logiciel

Algorithmes et programmes



- Algorithmes :
 - Des idées
 - Des mathématiques
- Programmes :
 - Des œuvres de l'esprit
 - Du discours
 - Humain → humain
 - Humain → ordinateur
- Traitements :
 - Lorsque exécutés

- Analogue au processus de création littéraire

Droit d'auteur adapté

- Par son rattachement au droit d'auteur, le logiciel est assimilé à une œuvre de l'esprit
 - En France, loi du 3 juillet 1985
 - Directive européenne 91/250/CE (1991)
 - Article 10 des accords ADPIC (1994)
 - Article 4 du traité OMPI WCT (1996)
- Le logiciel est cependant aussi un produit substituable voué à rendre un service
 - Adaptation du droit d'auteur
 - Nous parlerons donc de « Droit d'auteur adapté »
 - Question de la garantie

Adaptation des droits patrimoniaux (1)

- Ajout de nouvelles exceptions
 - « Actes nécessaires pour permettre l'utilisation du logiciel »
 - « Observation du fonctionnement »
 - En fait, simple rappel de la loi
 - Copie de sauvegarde
 - Si aucun autre moyen fourni par l'éditeur
 - La décompilation
 - Interdite sauf « à fin d'interopérabilité »
- Suppression de l'exception de copie privée
 - Nécessite une licence pour chaque exemplaire du logiciel

Adaptation des droits patrimoniaux (2)

- Pour les auteurs salariés ou agents publics (et leurs stagiaires dans la recherche), transfert automatique de la titularité des droits patrimoniaux à l'employeur
 - C'est l'employeur qui décide de la vie de l'œuvre :
 - Divulcation, choix de la licence, etc.
 - Apparition d'un statut d'« auteur prolétaire »
 - Déjà entamé avec la notion d'« œuvre collective »
 - Ne concerne que les logiciels réalisés dans le cadre professionnel, pendant le temps de travail !

Adaptation des droits patrimoniaux (3)

- Les auteurs non salariés restent titulaires des droits patrimoniaux sur leurs œuvres logicielles
 - Nécessité de transférer les droits des sous-traitants
 - Problème des stagiaires non rémunérés
 - Secteur privé ou public hors recherche

Adaptation des droits extra-patrimoniaux

- Amoindrissement des droits extra-patrimoniaux
 - L'auteur salarié ou fonctionnaire ne peut s'opposer à la modification de l'œuvre
 - Reste le droit au nom...
 - Plus courant dans le monde du jeu vidéo que de la comptabilité

Portée de la protection (1)

- Est couvert par le droit d'auteur adapté au logiciel tout ce qui relève de l'expression d'algorithmes mathématiques
 - Le code source et les codes objets qui peuvent en être dérivés
 - Expression formelle de ce que le logiciel fait
 - Exprime la personnalité de ses auteurs
 - Le « matériel de conception préparatoire »
 - Expression formelle de ce que le logiciel doit faire
 - A été considéré comme n'exprimant pas nécessairement d'originalité, donc jugé insusceptible de protection par le droit d'auteur classique

Portée de la protection (2)

- La documentation bénéficie d'un statut hybride
 - Aspects originaux couverts par le droit d'auteur classique
 - Aspects supposés potentiellement non originaux couverts par le droit d'auteur adapté
- Les éléments graphiques et sonores sont couverts par le droit d'auteur classique
 - De même qu'il existe des droits différents pour les auteurs d'un texte et de ses illustrations
 - Décision « Cryo », Civ. 1re, 25 juin 2009, n° 07-20.387

Organisation juridique et économique de la création logicielle

Car on peut (bien) vivre de son art...

Modèle économique de la création logicielle (1)

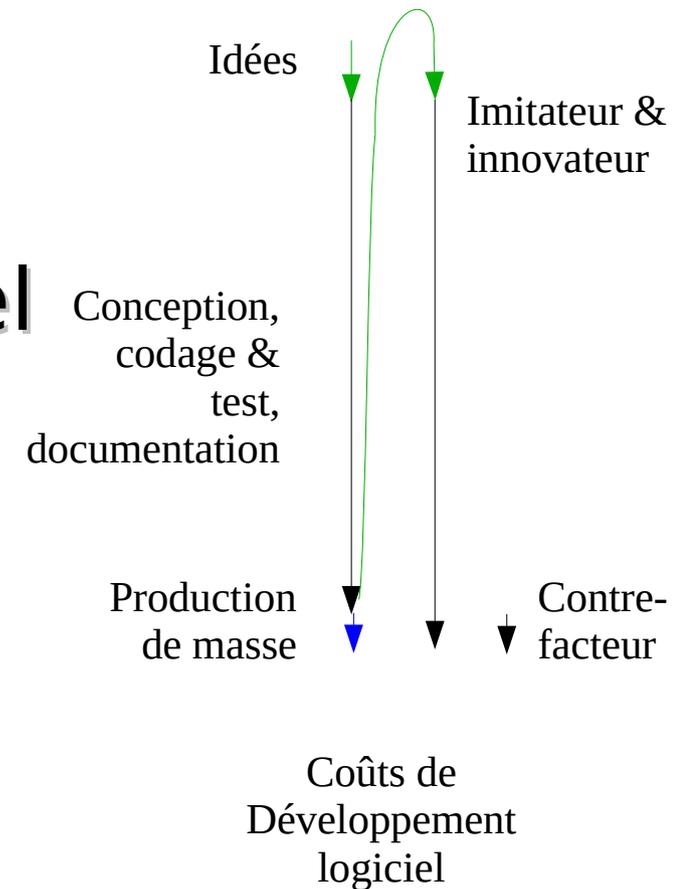
- La vision promue par la Commission européenne est parfaitement décrite dans les considérants de la directive 91/250/CE :
 - Secteur industriel de biens substituables
 - Nécessité d'établir les règles d'une concurrence libre et non faussée
 - Respect des droits des auteurs

Modèle économique de la création logicielle (2)

- L'adaptation des droits patrimoniaux vise à adapter les modalités d'exploitation économique aux spécificités de ce type d'œuvres
 - Droits individuels accordés au service d'une vision macro-économique
 - Tout comme le droit de suite est spécifique aux œuvres plastiques

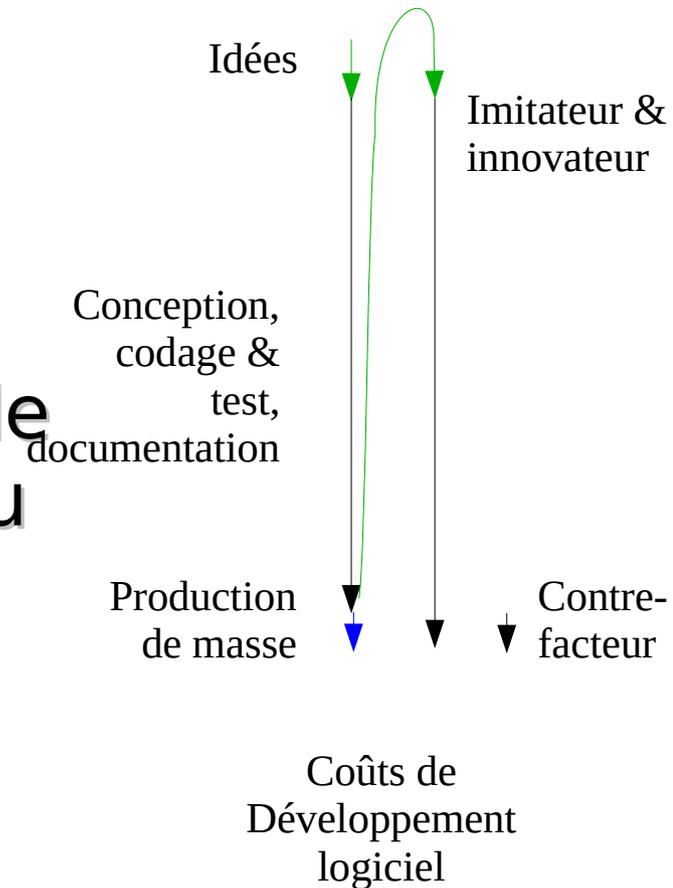
Modèle économique de la création logicielle (3)

- L'observation des fonctionnalités d'un logiciel existant permet à un nouvel arrivant de réaliser un logiciel concurrent
- Pendant ce temps, l'innovateur initial peut/doit progresser
 - S'il ne le fait pas, il sera dépassé par les innovations des autres



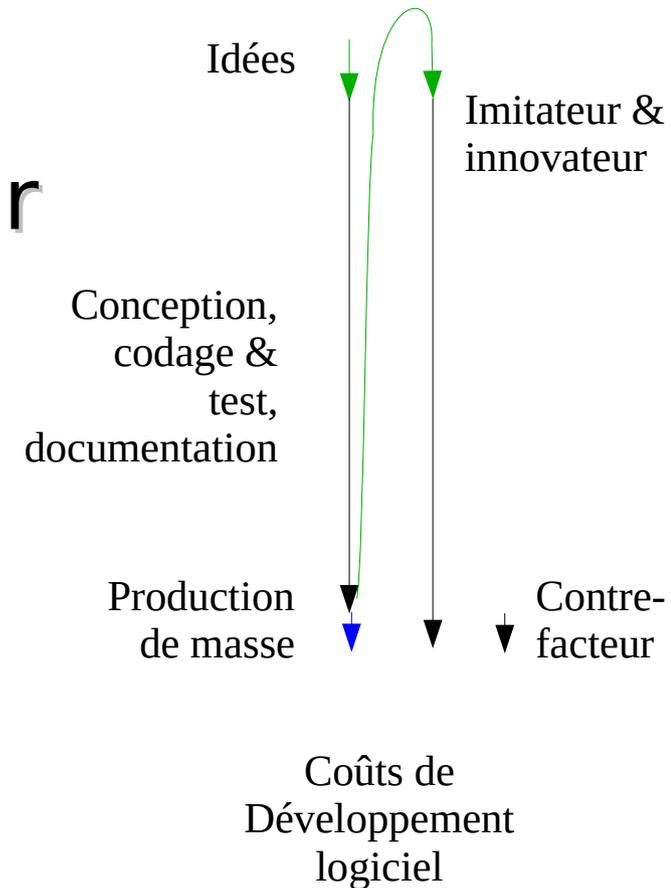
Modèle économique de la création logicielle (4)

- Il n'y a concurrence libre et non faussée que si le concurrent paye les mêmes coûts d'entrée sur le marché
 - Il ne peut donc réutiliser le code existant, par copie servile et/ou traduction et/ou décompilation
 - Doit re-coder intégralement
 - Le coût sera néanmoins souvent moindre, du fait de l'évolution des technologies logicielles



Modèle économique de la création logicielle (5)

- Nécessité de traiter spécifiquement le problème des marchés captifs créés par les formats de données
 - Dispositions relatives à la décompilation « à fin d'interopérabilité »



Compilation, décompilation et interopérabilité

Faire et défaire,
c'est toujours travailler !

Langages de bas et haut niveau (1)

- Les ordinateurs ne comprennent que le « langage machine »
 - Instructions très rudimentaires
 - Spécifique à un processeur donné
 - Et à un système d'exploitation donné
 - Code très peu expressif
 - Risques de bogues élevé
 - Proportionnels au nombre de lignes que l'on écrit

Langages de bas et haut niveau (2)

- Besoin d'écrire des programmes dans des langages plus expressifs, dits « de plus haut niveau »
 - Meilleure compréhension du code
 - Indépendance vis-à-vis du type de processeur
 - Meilleures maintenabilité et portabilité
 - Coût économique bien inférieur
- Des outils automatiques permettent la traduction de textes d'un langage vers un autre

Compilation et décompilation (1)

- La compilation consiste à traduire un programme écrit dans un langage de haut niveau en un programme fonctionnellement équivalent écrit dans un langage de bas niveau, susceptible d'être exécuté par un ordinateur
 - Le « code source » est le programme écrit en langage de haut niveau que l'on veut traduire
 - La « forme préférée » d'écriture d'un programme
 - Le « code objet » est le programme résultant écrit dans un langage de bas niveau
 - Définit le « programme exécutable »

Compilation et décompilation (2)

- On appelle « décompilation » l'action inverse de la compilation, permettant d'exprimer dans un langage de haut niveau un programme originellement écrit dans un langage de bas niveau
- Bien plus difficile à mettre en œuvre
 - En fait, théoriquement impossible sans ajout de connaissance depuis l'extérieur
 - Deuxième principe de la thermodynamique
 - Informations structurelles de haut niveau « diluées » dans le code objet

Traduction automatique (1)

- La traduction du code source en code objet se faisant de façon automatique :
 - Le code objet est une œuvre intégralement dérivée du code source
 - Aucun apport original d'un autre auteur
 - La sémantique du programme initial doit être intégralement respectée
 - L'apport est dans les idées qui rendent un traducteur automatique meilleur que ses concurrents
 - Pas de droits acquis par le créateur du traducteur sur les œuvres dérivées qu'il contribue à produire

Traduction automatique (2)

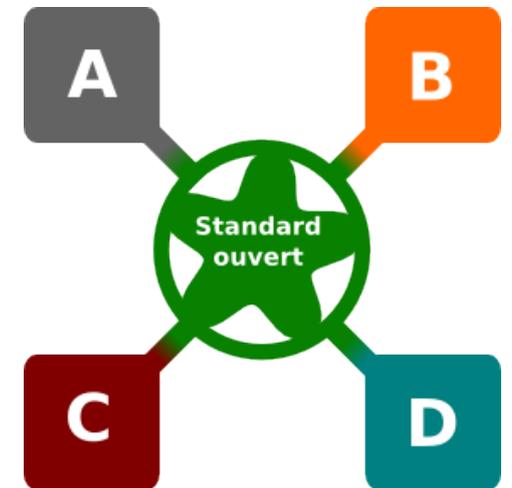
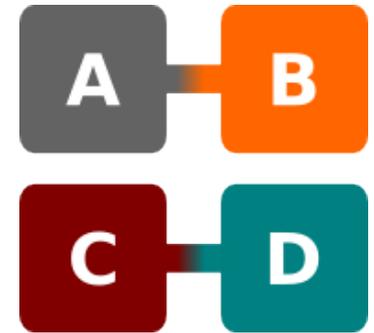
- Le créateur d'un outil n'a aucun droit sur les créations réalisées au moyen de cet outil
 - Pensez aux dictionnaires, pinceaux, etc...
- Cas particulier : inclusion de fragments de code issus du compilateur dans le programme exécutable
- Question des créations produites par les pseudo-« intelligences artificielles »
 - « Valladolid 2.0 » !

Interdiction de la décompilation

- Compiler et décompiler créent des œuvres dérivées de l'œuvre initiale
- Ces copies ne peuvent donc être exploitées qu'avec l'accord de l'ayant droit des œuvres initiales
- La décompilation est donc interdite, sauf permission explicitement donnée par l'ayant droit

Compatibilité n'est pas interopérabilité !

- Il y a compatibilité quand deux produits peuvent fonctionner ensemble
 - Accord contractuel entre les parties
 - L'entité contrôlant le format « compatible » étend son monopole
- Il y a interopérabilité quand deux produits peuvent fonctionner ensemble et que l'on sait pourquoi



Recherche de l'interopérabilité (1)

- Spécificité bienvenue du droit européen
 - Instaurée par la directive 91/250/CE
 - Maintenu par la directive 2009/24/CE
- La décompilation des logiciels est interdite sauf afin de rechercher l'interopérabilité (L.122-6-1 IV° CPI)

Recherche de l'interopérabilité (2)

- Encadrement strict par trois conditions :
 - Actes accomplis par une personne ayant le droit d'utiliser le logiciel ou mandatée à cette fin
 - Informations non déjà disponibles par ailleurs
 - Actes limités aux parties du logiciel nécessaires
- Les informations obtenues ne peuvent être utilisées à d'autres fins et « porter atteinte au droit d'auteur »
- Nécessité de documenter le processus afin de ne pas être accusé de contrefaçon

Statut des interfaces (1)

- La libre concurrence suppose qu'on puisse interfacier un nouveau logiciel avec un logiciel existant
 - Voire le remplacer en s'interfaçant avec les logiciels tiers avec lesquels ce logiciel interagissait
- Or, lorsqu'on veut s'interfacier avec un logiciel existant, il faut respecter ses interfaces
 - Décrites dans des fichiers d'en-tête, des fichiers de classes, etc.

Statut des interfaces (2)

- Pour s'interfacer, il faut donc recopier le format des interfaces et leur nommage
 - On peut prétendre qu'il n'y a pas d'originalité dans le code d'interfaçage (Oracle vs. Google)
 - Et qui plus est que la réutilisation de la forme n'est pas soumise aux droits de l'auteur, dans le but d'autoriser l'interopérabilité (91/250/CE)
- Une interface ne peut être protégée
 - Une interface donne l'accès à une extension d'un langage

Licences

Code is law
Code is poetry
Code is life !

Licence (1)

- La licence est une offre de contrat de la part du fournisseur, qui définit les conditions d'utilisation d'une œuvre
 - Le terme juridique exact est : « pollicitation »
- Basée sur le droit d'auteur ou le copyright
 - Convention de Berne de 1886
- Classiquement, une licence limite les droits d'usage d'une œuvre :
 - Interdiction de diffusion publique
 - Interdiction de reproduction, même partielle
 - ...

Licence (2)

- Ne pas confondre « licence » et « *license* »
- Aux États-Unis, la « *license* » est un texte relatif au droit d'auteur, donc de niveau fédéral
 - Interprétation uniforme, à la différence des « *contracts* »
- En droit français, une « licence » est un contrat qui permet de d'organiser l'application du droit
 - Par exemple : licence de marque
 - À rapprocher des contrats d'adhésion (CGU)

Licences de logiciels (1)

- Il existe de nombreux types de licences de logiciels :
 - « Licence privative » (« propriétaire ») : tous les droits sont réservés par leur titulaire
 - Cas de la majorité des logiciels sous rémunération
 - Dans la quasi totalité des cas, pour ces logiciels :
 - Le client n'est propriétaire que du support, et pas du logiciel qu'il contient
 - Le fournisseur dégage toute responsabilité en cas de vices cachés (bogues)
 - Dans les limites du droit applicable
 - Peu de sanctions pour vice caché
 - Le fournisseur peut arrêter la maintenance du logiciel à tout moment

Licences de logiciels (2)

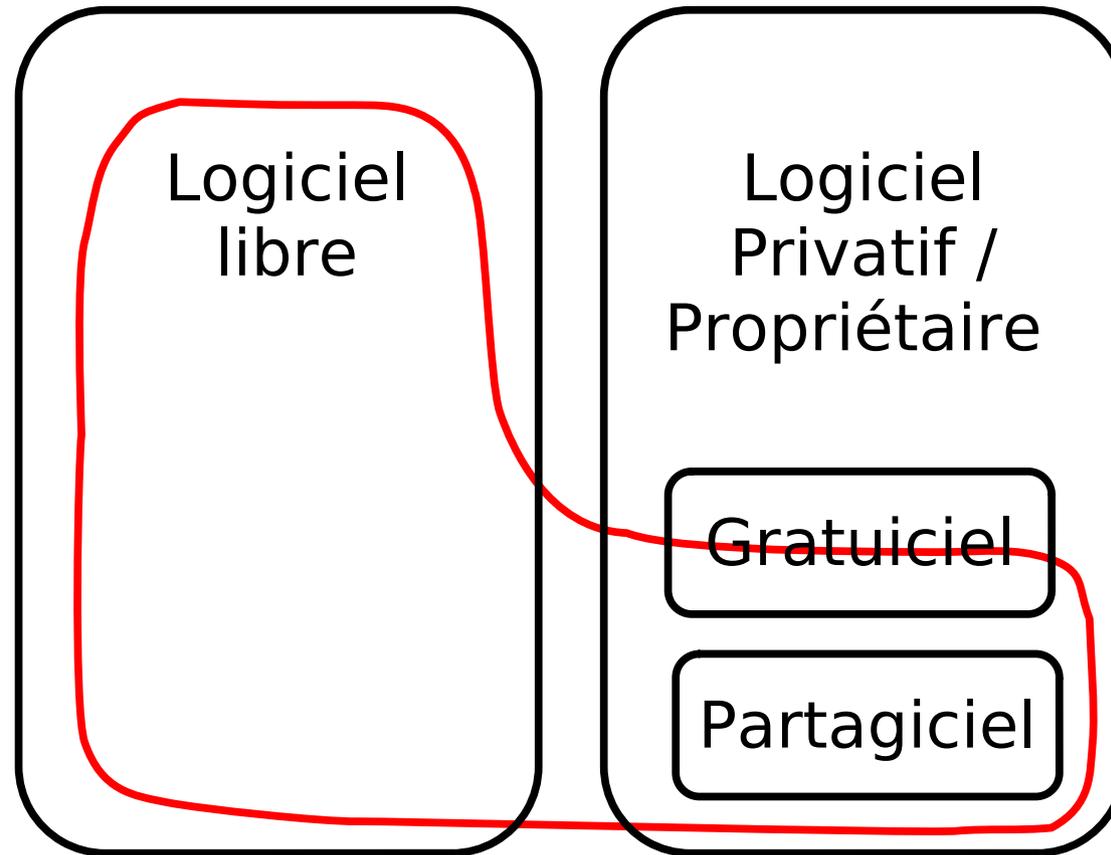
- « Licence partagicielle » (« *Shareware* ») :
licence privative autorisant la diffusion gratuite du logiciel mais pour lequel une contribution est demandée au bout d'une période d'essai
 - Licence privative où seul diffère le mode de distribution
- « Licence gratuicielle » (« *Freeware* ») :
licence privative autorisant la fourniture gratuite du logiciel mais ne donnant pas nécessairement d'autres droits
 - Parfois pas même celui de redistribution

Licences de logiciels (3)

- « Licence libre » (« *Free software* ») : licence donnant de nombreux droits aux utilisateurs
 - N'est pas équivalent à un « *freeware* » !
 - Pas nécessairement gratuit
- « De domaine public »
 - Domaine public volontaire

Récapitulatif des types de licences

Libre
téléchargement



D'après : <http://www.gnu.org/philosophy/categories.html>

Logiciel et licences libres

Free, Free,
Set them free (ouaouuah...)

Logiciel libre

- Le « logiciel libre » est une innovation juridique et non pas technique
 - S'appuie sur le mécanisme du droit d'auteur applicable aux logiciels
 - Garantit aux usagers des droits et des devoirs
- Permet l'émergence de modèles économiques déconcentrés adaptés à l'économie immatérielle
 - Annule le coût de transaction juridique
 - Coût de transaction technique (coût marginal de copie) rendu nul par l'Internet

Licences libres (1)

- Ont en commun les « quatre libertés »
 - Liberté d'utilisation pour tout usage
 - Liberté de copie du logiciel obtenu
 - Liberté de modification du logiciel
 - Nécessite l'accès au code source
 - Liberté de redistribution du logiciel modifié
 - Capitalisation du savoir
 - Mutualisation des développements
 - C'est une permission, pas une obligation !

Licences libres (2)

- Les différences entre licences libres portent sur les modalités de redistribution du logiciel modifié :
 - Si persistance des obligations de redistribution du code source : « *copyleft* »
 - Liberté collective
 - Si absence de cette obligation : « *non copyleft* », ou « *permissive* »
 - Liberté individuelle

« Libre » c/ « à sources ouverts » (1)

- Un certain nombre d'entrepreneurs précurseurs de la filière libre dans la *Silicon Valley* se sont inquiétés que le terme « *free* » pouvait être perçu comme « anti-business »
- Création :
 - Du terme « *open source software* »
 - De l'« *Open Source Initiative* »
 - De l'« *Open Source Definition* »

« Libre » c/ « à sources ouverts » (2)

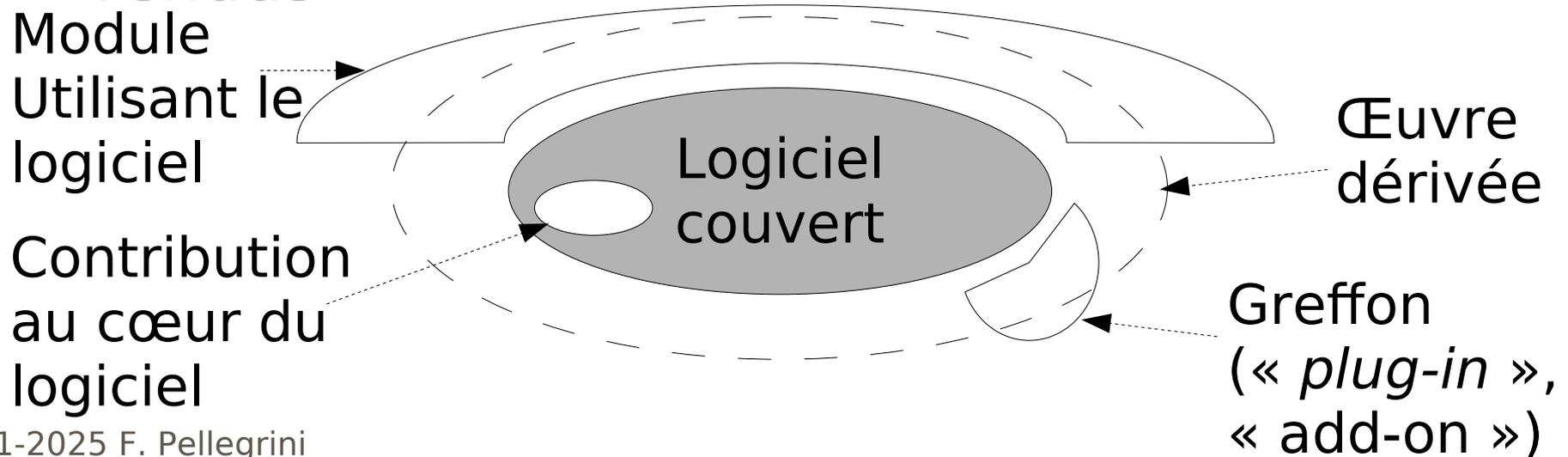
- Open Source Definition :
 - Libre redistribution du code original
 - Accès au code source
 - Libre redistribution des œuvres dérivées
 - Intégrité du code source initial (vs. patches)
 - Pas de discrimination contre les personnes/groupes
 - Pas de discrimination contre les domaines
 - Aucune licence supplémentaire
 - Licence non restrictive à un produit
 - Licence non restrictive à d'autres produits
 - Licence technologiquement neutre

« Libre » c/ « à sources ouverts » (3)

- Recouvrement presque intégral entre les périmètre « libre » et « *open source* »
 - Existence de quelques licences « *open source* » mais pas considérées comme « libres »
 - Problème maintenant résolu
- Problème de dévoiement du terme « *open source* »
 - Et « *open* » en général
- Utilisation préférentielle du terme « *libre* » en français

Mode d'action des licences libres (1)

- Interaction avec un module logiciel :
 - Utilisation : dans un autre logiciel ou par interaction
 - Modification : travail « au cœur du logiciel »
 - Création d'un greffon : interaction par une interface (API) dédiée pour étendre les services rendus



Mode d'action des licences libres (2)

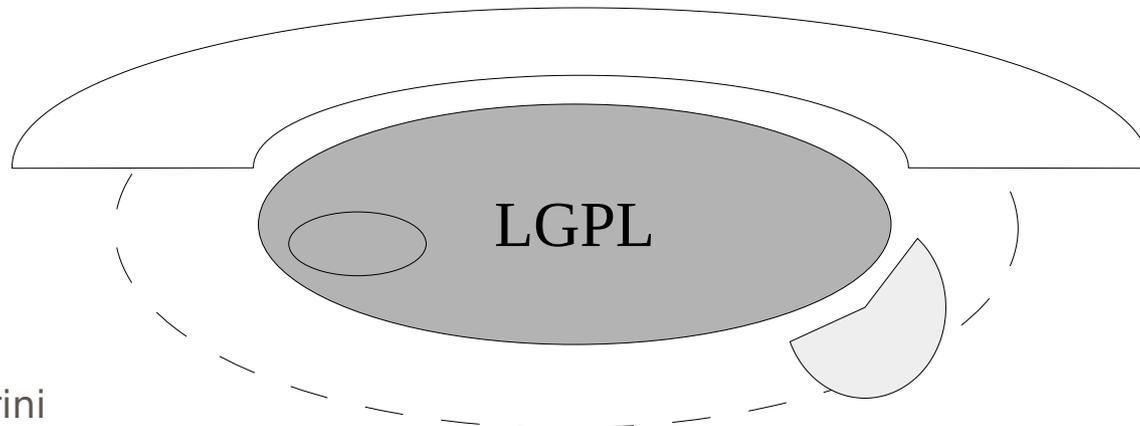
- Les licences libres veillent à ce que « leurs termes » s'appliquent à toute œuvre dérivée
 - Permet de réguler l'usage qui est fait du module
 - Permet de réguler le choix de la licence des modules liés au module couvert
- N'impose pas toujours que la licence elle-même doive couvrir les autres éléments constitutifs de l'œuvre dérivée et celle-ci !
 - Dépend des modalités de la licence

Principaux types de licences libres

- Trois principaux types de licences libres :
 - Licences « persistantes »
 - Licences « évanescentes »
 - Licences « diffusives »

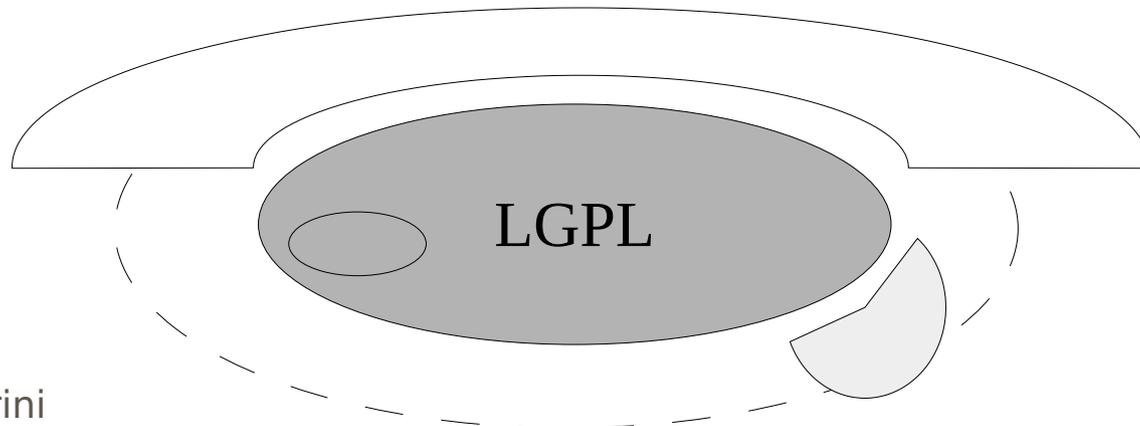
Licences « persistantes » (1)

- Aussi appelées : « pérennes », « à *copyleft* faible » (« *weak copyleft* »)
- Exemples : LGPL, CeCILL-C



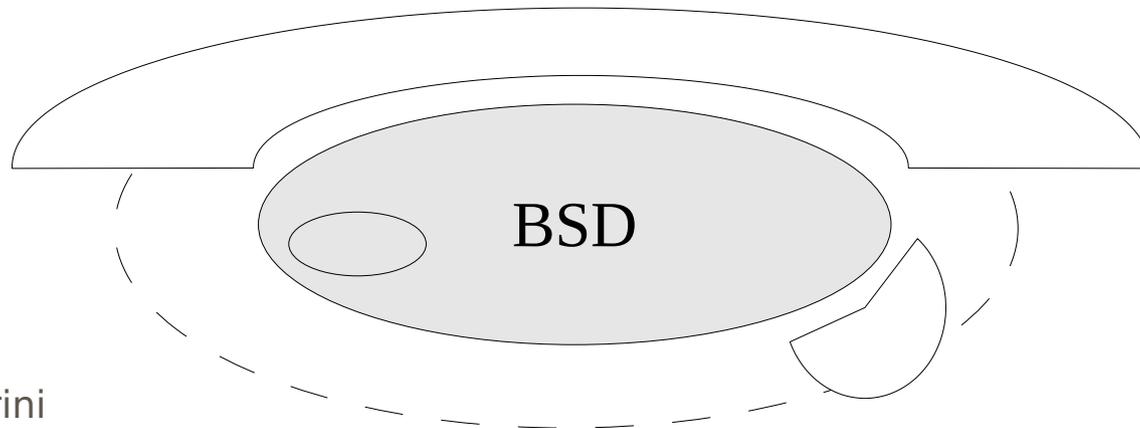
Licences « persistantes » (2)

- Le module peut être combiné à du code sous licence non libre et l'œuvre dérivée peut être diffusée sous licence non libre
- Le code source des versions modifiées doit être rediffusé lorsque celles-ci le sont
 - Ce qui était libre le reste



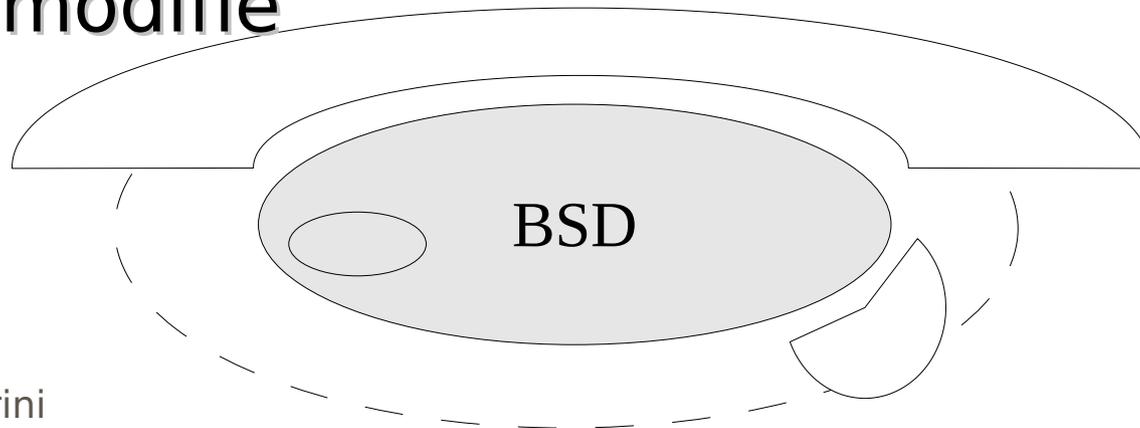
Licences « évanescentes » (1)

- Aussi appelées : « permissives », « non copyleftées » (« *non copyleft* »)
- Exemples : BSD, CeCILL-B



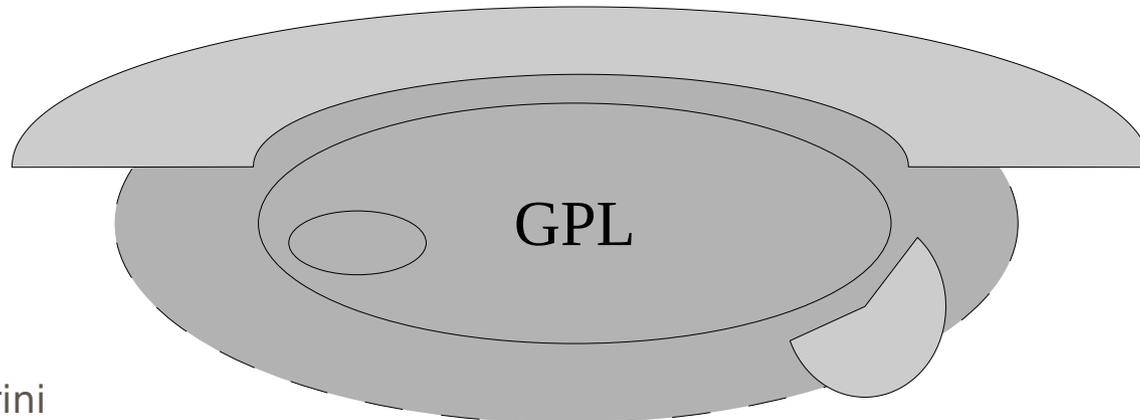
Licences « évanescentes » (2)

- Le module peut être combiné à du code sous licence non libre et l'œuvre dérivée peut être diffusée sous licence non libre
- Pas d'obligation de rediffuser le code source du module
 - Possibilité de « refermer » du code libre
 - La CeCILL-B permet de changer la licence du code modifié



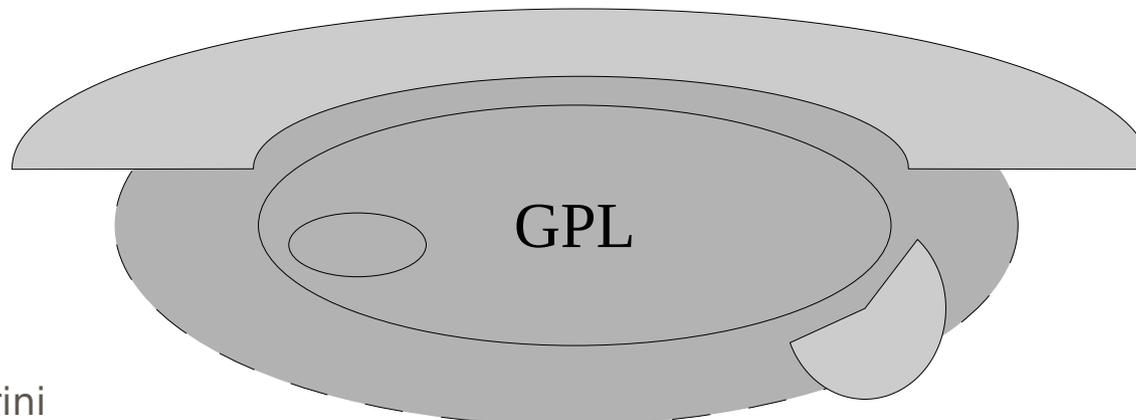
Licences « diffusives » (1)

- Aussi appelées : « à copyleft fort »
(« *strong copyleft* »)
 - Les opposants au logiciel libre disent parfois « contaminantes »
 - Terme non neutre, à éviter !
- Exemples : GPL, CeCILL(-A)



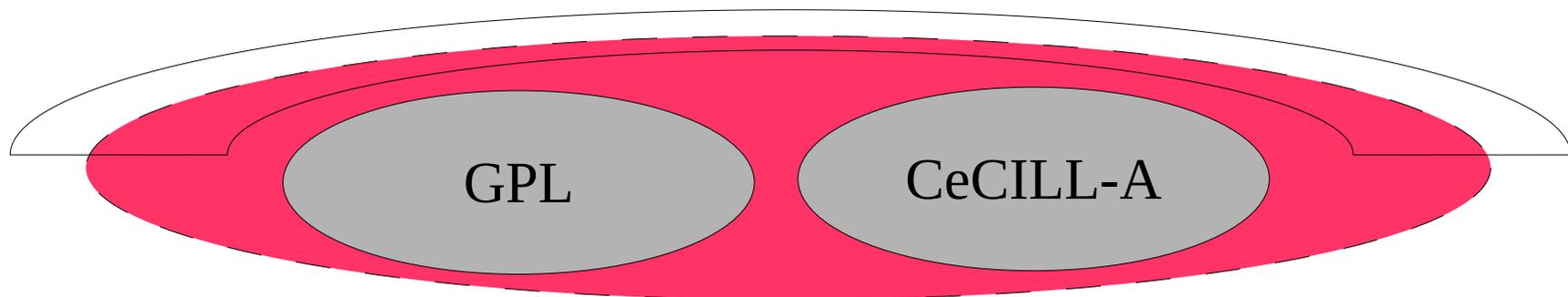
Licences « diffusives » (2)

- L'œuvre dérivée doit être distribuée selon les « termes » de la licence du module
 - Pas nécessairement la même licence !
- Les modules fortement liés doivent donc être couverts par des licences libres compatibles avec les termes de la licence du module



Licences « diffusives » (3)

- Les licences diffusives peuvent induire des conflits juridiques lorsqu'on cherche à lier ensemble deux modules sous licences diffusives différentes



- Certaines licences « mineures » possèdent des clauses permettant de résoudre ces conflits
 - La CeCILL-A cède explicitement le pas à la GPL

Multi-licenciage

- L'ayant droit d'un logiciel peut choisir de diffuser celui-ci avec le type de licence de son choix
- Il peut même diffuser le même code source, par plusieurs canaux différents, avec des licences différentes
 - Politique de licences multiples : « *dual licensing* »
 - Attention au suivi des versions et contributions successives !

Déclenchement des licences (1)

- Les premières licences libres ont été créées en un temps où les usages des réseaux étaient peu développés
 - Mise en œuvre centralisée des logiciels par leurs usagers (personne physique ou morale)
- Ces licences sont déclenchées par la réception d'un exemplaire du logiciel
 - Donne les droits d'accès au code source et impose les règles applicables à la redistribution
 - GPL, LGPL, etc.

Déclenchement des licences (2)

- Le développement des systèmes de partage de fichiers pair-à-pair fait que des personnes peuvent être redistributrices de logiciels sans avoir décidé d'accepter la licence
- Ces licences ont dû être amendées pour tenir compte de réceptions involontaires, comme c'est le cas avec les réseaux pair-à-pair
 - Différence entre « *conveying* » et « *(ancillary) propagating* » (art. 9 GPL v3)

Déclenchement des licences (3)

- L'émergence des usages logiciels en mode SaaS (« *Software as a Service* ») fait que de nombreuses personnes utilisent des logiciels libres sans pouvoir bénéficier des termes de leurs licences
 - L'utilisation à distance ne constitue pas une réception du logiciel
- De nouvelles formes de licences ont été créées afin de remédier (partiellement) à ce problème : les licences déclenchées par l'utilisation

Déclenchement des licences (4)

- Pour autant, le déclenchement par l'usage ne résout pas réellement les problèmes posés par le passage à un environnement « SaaS »
 - Le fait de recevoir copie du logiciel et de modifier cette copie n'oblige pas le prestataire SaaS à modifier sa propre copie
 - Un logiciel utilisé en SaaS et ne permettant pas l'exportation des données laissera l'utilisateur dans l'impossibilité effective de changer de fournisseur

Cas d'usage des licences libres

Cas type : libre téléchargement (1)

- Logiciel dont le marché est très large, constitué d'entités non concurrentes ou dont ce n'est pas le cœur de métier
 - Cas des bibliothèques et logiciels de service
- Possibilité de créer une communauté d'utilisateurs et de contributeurs
 - Les libertés d'usage augmentent sa taille et sa valeur
- Mutualisation de la maintenance et des développements ultérieurs

Cas type : libre téléchargement (2)

- Diffusion sous licences libres persistantes ou diffusives

Cas type : consortium fermé (1)

- Logiciel métier dont le marché est étroit
- Mutualisation de l'effort de développement entre les membres du consortium
 - Titularité des droits au prorata des apports
 - Totale liberté d'associer de nouveaux membres à des évolutions futures du travail réalisé en commun
 - Aucun désir de donner gratuitement aux concurrents ce qui a coûté à produire
 - Libre ne veut pas dire gratuit !
 - Libre ne veut pas dire « en libre téléchargement »

Cas type : consortium fermé (2)

- Diffusion sous licences libres persistantes ou diffusives

Cas type : partenariat privilégié (1)

- Logiciel ou bibliothèque métier potentiellement utilisable par une communauté plus large
- Choix d'un partenaire privilégié fournissant un retour sur expérience

Cas type : partenariat privilégié (2)

- Distribution double sous licences libres :
 - Fourniture au partenaire sous licence évanescence pour lui permettre l'inclusion du logiciel au sein de produits dont les caractéristiques sont cachées aux concurrents
 - Mise en libre accès sous licence diffusive pour contributions de la communauté et la réalisation éventuelle de logiciels analogues mais au code source accessible à tous

Pourquoi développer sous licences libres

- Outils idéaux pour la préservation du patrimoine intellectuel
- Coût nul
 - L'ajout des mentions de licences dans chaque fichier source est suffisant
 - Le dépôt à des organismes de type APP (pour un coût dérisoire) apporte une preuve d'antériorité
- Mutualisation de l'effort de développement
- Meilleure pérennité pour les clients
 - Devient un argument commercial

Bonnes pratiques de développement

Mieux vaut prévenir que guérir...

Problématique juridico-technique

- L'analyse juridique doit être concomitante à l'analyse technique
 - Le choix des composants logiciels et de leurs licences conditionne le coût et les modèles économiques du logiciel réalisé
 - Mieux vaut gérer les problèmes en amont qu'en aval
- La compatibilité entre licences doit être étudiée de façon fine
 - Pas nécessairement simple
 - Des logiciels dédiés peuvent (partiellement) automatiser ce processus

Mise en œuvre pratique

- Utiliser des gestionnaires de version pour tracer l'intégralité des contributions
 - Permet de garantir (semi-)automatiquement le respect du droit au nom
 - Permet de savoir quoi re-coder en cas d'incertitude juridique
- Utiliser deux gestionnaires de version :
 - Un gestionnaire privé pour les personnes dont on connaît (et maîtrise) la titularité des droits
 - Un gestionnaire public éventuel pour susciter les contributions de la communauté

Les « brevets logiciels »

On pourrait en rire,
si ce n'était à en pleurer...

Portée du droit d'auteur adapté

- Garantit qu'on ne peut exploiter un programme en dehors des clauses stipulées par sa licence
- N'interdit pas d'écrire un nouveau programme :
 - Aux fonctionnalités similaires
 - Compatible au niveau des formats d'entrée/sortie
 - Interopérable avec le programme original

Protection du logiciel en Europe

- Les programmes informatiques sont protégés par le droit d'auteur
 - Directive 91/250/CE puis 2009/24/CE
- Les logiciels sont explicitement exclus du champ de la brevetabilité par l'article 52§2c de la Convention du brevet européen de Munich (1973), tout comme les jeux et les mathématiques

Brevets et logiciels (1)

- Néanmoins, un procédé physique brevetable ne doit pas être exclu de la brevetabilité parce qu'il embarque un logiciel
 - Prise en compte de la numérisation du contrôle des procédés industriels
- Exclusion de brevetabilité limitée au logiciel « en tant que tel »
- Permet de revendiquer des inventions « pilotées par ordinateur »
 - Mais l'innovation ne réside pas dans le logiciel

Brevets et logiciels (2)

- Avec le basculement de l'économie vers le logiciel, les offices de brevets ont vu des pans entiers de l'économie leur échapper
- Tentation de capter à leur profit une partie de cette manne
 - Extension du système de brevets au logiciel
 - Contraire à la volonté du législateur

Brevets et logiciels (3)

- Nécessité de tordre la loi, au moyen de deux impostures intellectuelles majeures :
 - Extension abusive du périmètre de brevetabilité
 - Confusion de la fin et des moyens

Critères de brevetabilité

- Pour être brevetable, une innovation doit :
 - Appartenir à un domaine brevetable
 - Domaine « technique », impliquant le contrôle des « forces de la nature »
 - Les pratiques médicales étaient, jusqu'il y a peu, exclues de la brevetabilité
 - Être nouvelle
 - Pas de divulgation préalable
 - Démontrer une activité inventive
 - « Non évidente à "l'homme de l'art" »
 - Censé éviter les brevets triviaux
 - Être susceptible « d'application industrielle »
 - Censé limiter le périmètre de la brevetabilité

Extension abusive (1)

- Changement arbitraire des règles d'examen :
 - Anciennes règles : la « nouveauté » ne devait concerner que les revendications « techniques »
 - Nouvelles règles : « nouveauté » considérée « dans son ensemble », donc même si elle ne concerne que les caractéristiques « non techniques »

Extension abusive (2)

- Postulat d'existence de logiciels « non en tant que tels » susceptibles d'être brevetés
 - Permet de revendiquer un logiciel nouveau (non « technique ») s'exécutant sur un ordinateur (« technique » mais non nouveau)

Extension abusive (3)

- Terme novlangue d'« invention mise en œuvre par ordinateur »
 - Ce que met en œuvre un ordinateur, c'est un algorithme !
 - Au moyen d'un logiciel qui exprime cet algorithme en tant que création de forme
 - Prétend qu'un algorithme peut être une « invention » au sens de la brevetabilité
 - Alors que les mathématiques ne sont pas brevetables « en tant que telles »

Confusion fins / moyens (1)

- Assertion erronée selon laquelle les logiciels auraient un « effet technique »
 - C'est-à-dire une action dans le monde physique
- Un logiciel manipule des valeurs symboliques
 - Décorrélé de toute réalité physique
 - Exécutable « à la main » ou dans un simulateur sans produire de tels « effets techniques »
- Ce sont les périphériques de l'ordinateur qui interagissent avec le monde physique

Confusion fins / moyens (2)

- L'amélioration d'un logiciel n'est pas une « solution technique » à un « problème technique »
 - Le logiciel ne modifie pas la nature de l'interaction entre l'ordinateur, ses périphériques et leur environnement
- Dans le cas contraire, un pilote de F1 serait brevetable parce qu'il améliore le temps de parcours d'un circuit (« problème technique ») avec une voiture (« moyen technique »)

Confusion fins / moyens (3)

- Changement arbitraire des règles d'examen :
 - Anciennes règles : proposer une « solution technique » à un « problème technique »
 - Nouvelles règles : simple usage de « moyens techniques » pour résoudre n'importe quel type de problème, y compris « non technique »
- Permet de revendiquer les « *business methods* », algorithmes du monde réel :
 - Méthodes de commerce en ligne
 - Calcul de montants de pensions au moyen d'un ordinateur

Portée du brevet algorithmique (1)

- Les brevets algorithmiques ne concernent pas directement les programmes
- Ils visent à monopoliser les concepts sous-jacents, tels que :
 - Ce qu'un programme fait :
 - Quel problème concret ce programme résout, c'est-à-dire quelle « *business method* » il met en œuvre
 - Comment il le fait, et plus particulièrement :
 - Quelles données d'entrée il accepte
 - Quelles données de sortie il produit
 - Comment il interagit avec d'autres programmes

Portée du brevet algorithmique (2)

- Ces brevets portent sur :
 - Les « *business methods* » (« algorithmes du monde réel »)
 - Étapes suivies par un utilisateur achetant des biens sur Internet (brevet « 1-Click » d'Amazon), ...
 - Les formats de fichiers
 - Documents, feuilles de calcul, image (GIF), son (MP3), ...
 - Les protocoles de communication entre programmes
 - Les algorithmes
 - Chiffrement des données, ...
- Concepts logiquement indissociables

Exemples de brevets délivrés par l'OEB bien qu'invalides dans l'Union européenne

L'Urgence du E-Commerce Européen

http://webshop.ffii.org/

1 **Votre boutique web est BREVETÉE!**

4 Musiques **2** **NOUVEAUTÉ: COMMANDEZ PAR TÉLÉPHONE PORTABLE!** **15** Obtenez de l'aide directement depuis nos bases de données de support internes! **?**

4 Films **3** Ajoutez au panier **7** Vous aimez ce résultat de recherche? Vous aimerez sans doute ceux-ci:

17 **15** Les coccinelles sont des insectes très utiles. Elles se débarrassent des parasites. Cependant les services contentieux des brevets logiciels sont en général bien trop gros pour elles. **8** Achetez la musique originale (mp3) **6** Achetez le film **19** <Entrez votre réduction> **18** Vous aimez ce résultat de recherche? Vous aimerez sans doute ceux-ci:

16 **5** Cliquez ici pour un plus grand aperçu **20** Rendez-vous dans l'un de nos magasins et composez/gravez votre propre DVD à la carte! **9** Paiement par carte de crédit **11** Demandez un prêt **10** Envoyez comme cadeau **12** **14** **13** Oui, je veux recevoir vos offres!

16 Aperçu de quelques chapitres: cliquez dans la télé ci-dessus!

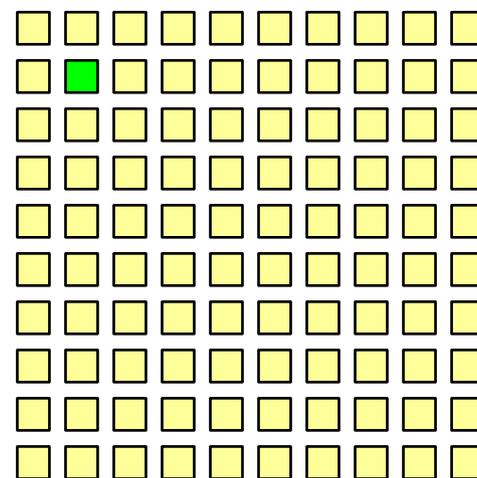
13 **14** **12** **10** **11** **18** **7** **6** **8** **15** **4** **3** **2** **1** **17** **16** **15** **14** **13** **12** **11** **10** **9** **8** **7** **6** **5** **4** **3** **2** **1**

Si nous n'avons pas votre commande en stock, elle sera immédiatement transmise à l'un de nos vendeurs affiliés!

Économie du « brevet logiciel » (1)

- Rêves de fortune

- Mon « invention » brevetée
- Techniques de programmation évidentes

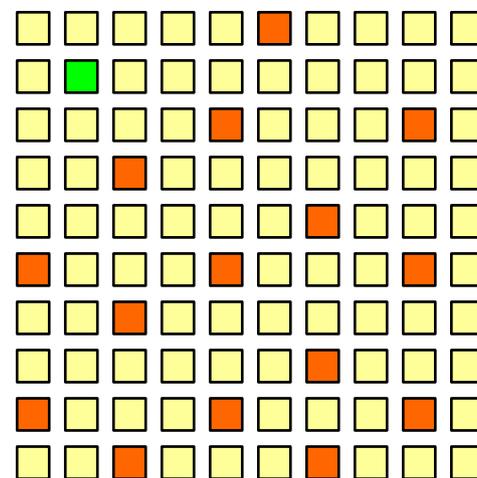


$$\text{Profit} = \text{Ventes} - \text{Develop} + \underbrace{\text{Revenu (■)} - \text{Coût (■)}}_{> 0}$$

Économie du « brevet logiciel » (2)

- Batailles de brevets logiciels

- Mon « invention » brevetée
- Techniques non brevetées
- Risque de poursuite pour contrefaçon de brevet



$$\text{Profit} = \text{Ventes} - \text{Develop} + \underbrace{\text{Revenu (■)} - \text{Coût (■)} - \text{Coût (■)}}_{< 0}$$

Économie du « brevet logiciel » (3)

$$\text{Profit} = \text{Ventes} - \text{Develop} + \underbrace{\text{Revenu (■)} - \text{Coût (■)} - \text{Coût (■)}}_{\text{En moyenne} < 0}$$

Petit éditeur :

Doit payer

$$\underbrace{\text{Revenu (■)} - \text{Coût (■)}}_{< 0} - \underbrace{\text{Coût (■)}}_{< 0}$$

Gros éditeur :

Neutralise les brevets concurrents

$$\underbrace{\text{Revenu (■)} - \text{Coût (■)}}_{< 0} - \underbrace{\text{Coût (■)}}_{\approx 0}$$

Fonds de brevets :

Vit sur le dos du système

$$\underbrace{\text{Revenu (■)} - \text{Coût (■)}}_{\text{En principe} > 0} - \underbrace{\text{Coût (■)}}_{= 0}$$

Qui survit ?

- Les grands groupes
 - Échanges de portefeuilles de brevets
 - Position de quasi-monopole
- Les fonds de brevets
- Les petites entreprises désirant se faire racheter
 - Leurs brevets augmentent leur valeur supposée
 - Mais risque systémique ignoré !
- Les avocats, les experts en propriété industrielle, les offices de brevets
 - Consommement entre 10 et 40 % des moyens

Qui déperit ?

- Les petits éditeurs de logiciels, en général peu préparés aux batailles juridiques, par :
 - Manque de fonds (coût moyen d'un procès > 5 M\$)
 - Manque de portefeuilles de brevets à échanger
- Les entités utilisant du logiciel « maison »
 - Même si ces entreprises n'appartiennent pas au monde du logiciel
 - Même si elles ne les commercialisent pas
- Les développeurs de logiciels libres
 - Code source mis à disposition de la partie

adverse

Menaces sur l'innovation et l'interopérabilité

- Si le détenteur d'un brevet algorithmique refuse de concéder des licences :
 - Il est illégal d'écrire des programmes qui lisent ou produisent des formats de données brevetés
 - Les utilisateurs ne peuvent plus basculer vers d'autres produits pour traiter leurs données existantes
 - L'entrée sur le marché de nouveaux acteurs est réduite
 - Le choix des produits et des fournisseurs est réduit

Bilan du brevet algorithmique

- Instrument conçu pour l'industrie matérielle et étendu abusivement aux méthodes intellectuelles
 - Monopolise les algorithmes (= mathématiques !)
- Illégal en Europe mais accordé par les offices, qui vivent des annuités
- Contresens économique
 - Instrument anti-concurrentiel
 - Coûteux à obtenir et à maintenir
 - Menace le retour sur investissement
 - Favorise les délocalisations

Conclusion

Conclusion

- Les informaticiens ne doivent pas méconnaître le droit
 - Inclusion de paramètres juridiques dans tous les critères de décision, à toutes les étapes du cycle de vie d'un projet
- Les juristes ne doivent pas non plus méconnaître l'informatique
 - Pertinence des décisions de justice
 - Aberrations telles que le brevet algorithmique

Bibliographie – Vue d'ensemble

- *Histoires et cultures du Libre*

C. Paloque-Berges & C. Masutti

<https://archives.framabook.org/histoiresetculturesdulibre/>

Bibliographie – Droit (1)

- *Fondamentaux juridiques - Collaboration industrielle et innovation ouverte*

P. Moreau, C. Moulin, J. Pappalardo et
F. Pellegrini

https://cnll.fr/media/LivretBleu_Juridique-2eEdition_GT-

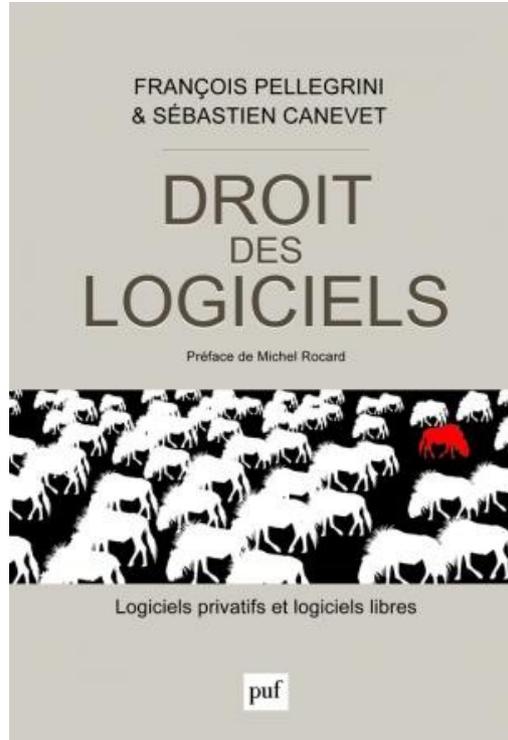
[LogicielLibre_Systematic_Nov2016_web.pdf](https://cnll.fr/media/LivretBleu_Juridique-2eEdition_GT-LogicielLibre_Systematic_Nov2016_web.pdf)

Bibliographie – Droit (2)

- *Droit des logiciels – Logiciels privatifs et logiciels libres*

F. Pellegrini & S. Canévet

https://www.puf.com/content/Droit_des_logiciels



Bibliographie – Droit (3)

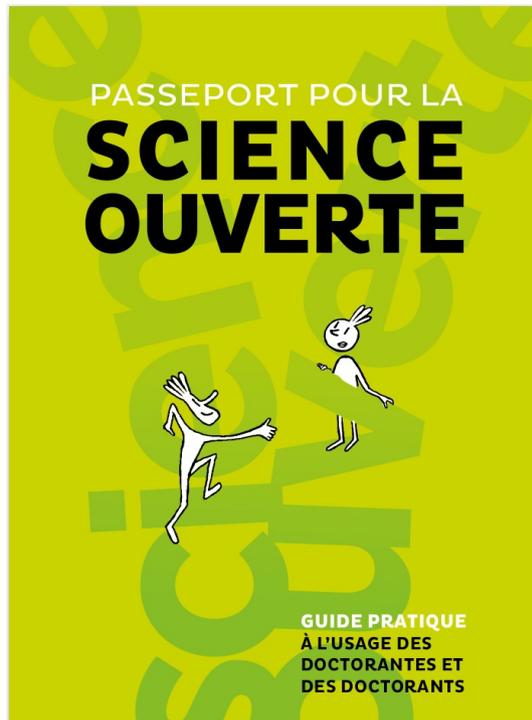
- *Option Libre – Du bon usage des licences libres*

B. Jean

<https://archives.framabook.org/optionlibre-dubonusagedeslicenceslibres/>

Bibliographie – Pratique

- *Codes et logiciels [Passeports S.O.]*
F. Pellegrini, R. Di Cosmo, L. Romary et al.
<https://www.ouvrirlascience.fr/science-ouverte-codes-et-logiciels/>



Bibliographie – Économie

- *Économie du logiciel libre*
 - F. Élie
<https://www.eyrolles.com/Informatique/Livre/economie-du-logiciel-libre-9782212124637/>